

Semantic Forms: What's
new, what's still missing?

About us

- Yaron Koren
 - Founder/CEO of WikiWorks
 - MediaWiki extension developer
 - Author of *Working with MediaWiki*
- Himeshi De Silva
 - Former Google Summer of Code student
 - Student from University of Moratuwa, Sri Lanka.

Section handling in Semantic Forms

- Since the beginning Semantic Forms has been a system only for editing template calls
- With the new 'section' tag from SF version 2.6, page sections too can be edited – not just templates
- Adds more flexibility and allows the defining of complete page structures

The 'section' tag

- Specifies a page section
- Creates a textarea in the form to populate the page section
- Definition,
 - {{{section | Plot}}}
 - This would create a textarea in the form where the 'Plot' section of the page is to be entered
- Special:CreateForm can be used to define sections

Parameters

- Allowed parameters are,
 - level=
 - mandatory
 - restricted
 - hidden
 - class=
 - rows=
 - cols=
 - autogrow
- Except for 'level' all other parameters are the same for textarea input

Demo

Unit tests

- First unit test scripts for Semantic Forms
- To run unit tests for the Semantic Forms extension
 1. Install PHPUnit
 2. From the MW root directory, run

```
php tests/phpunit/phpunit.php extensions/SemanticForms/tests
```

- Only covers {{{section}}} tag in SFFormPrinter class.

Section handling in Page Schemas

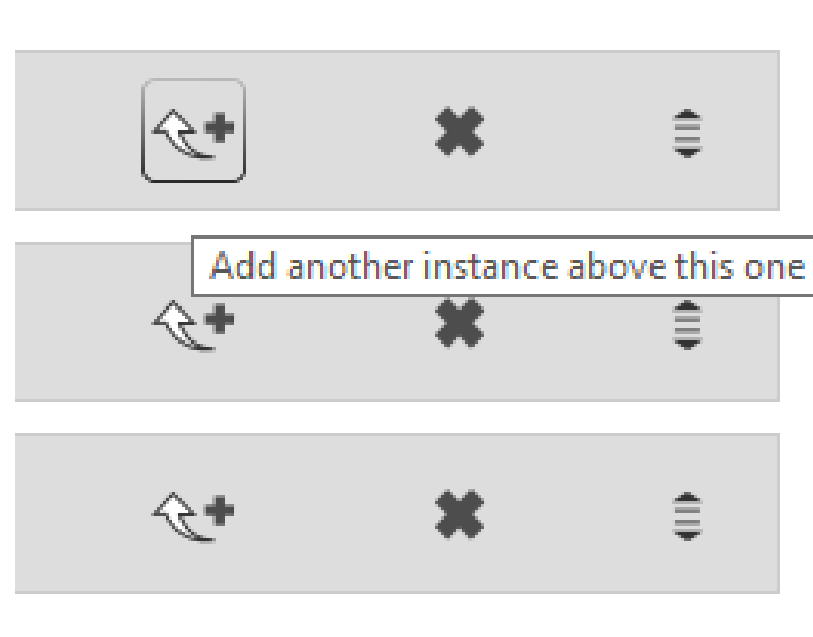
- The new <Section> XML element can be used to define sections in the schema

```
<Section name="Plot" level="2">
```

- The Special:EditSchema page can be used to add sections to the schema and modify their attributes.
- As before "Generate pages" will create the form with the templates and sections in the schema.

The "add another instance above" button

- Helpful when there are a large number of instances.
- Led to a streamlining of the multiple-instance template interface.



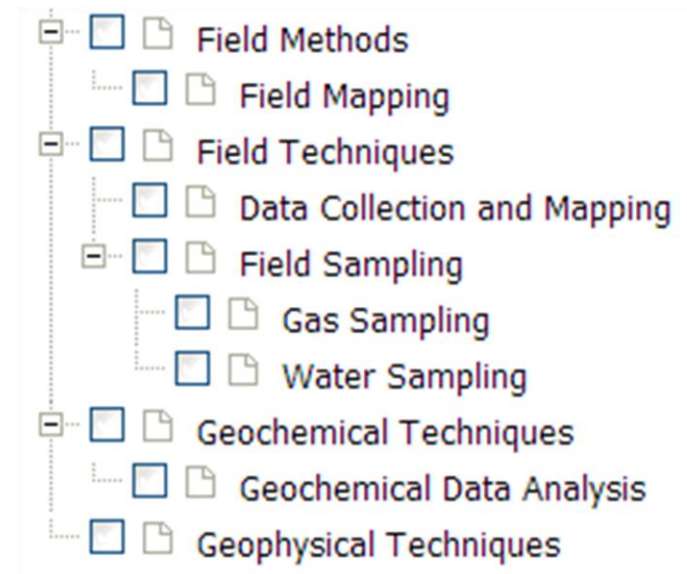
Native handling for the category input types

- "category" and "categories" used to rely on the CategoryTree extension; now they are handled natively, using the DynaTree JS library.

Old appearance (from openei.org):



New appearance, for same layout:



CKEditor/FCKeditor/"WYSIWYG" support for textareas removed

- These were three related WYSIWYG extensions.
- The first two are no longer maintained, but the third is, sort of - maybe support should have been kept for a little longer.

What's missing?

Missing feature: VisualEditor support

- VisualEditor = WYSIWYG editor developed for use on Wikipedia.
- In theory, much better than any previous WYSIWYG solution.
- Still not fully ready for widespread usage, though.

(Possibly) missing feature: Scribunto support

- Lets you define "modules", written in Lua - a substitute for #if, etc.
- Now in use on some Wikipedia templates
(see <http://en.wikipedia.org/wiki/Module:InfoboxImage>)
- Is anyone else using this thing? No one has asked about SF/SMW supporting it. (Whatever it means to support it.)

Missing feature: display aliases

- The user selects between "Alice", "Bob" and "Charlie", but what shows up in the page wikitext is "Employee 123", "Employee 456" or "Employee 789".
- Especially useful for pages with automatically-generated page titles, but has other uses, like adding an explanation of each value .

Missing feature: spreadsheet-style editing

- A JS-based spreadsheet-style editor, usable for editing multiple-instance templates.
- Might be tricky to implement.

How to get involved?

- Best communication methods: email (personal or mailing list), SF extension talk page.
- If I (Yaron) don't respond, feel free to keep pestering me.
- Contact Yaron if:
 - You want to add a feature
 - You want to fix a bug
 - You want to sponsor development on a feature/bug fix
 - You already have a code patch
 - You have an idea for the next Google Summer of Code. :)

Questions/comments